CHAPTER 7

INTERRUPT CONTROL UNIT

7.1 OVERVIEW

The Am186EM and Am188EM microcontrollers can receive interrupt requests from a variety of sources, both internal and external. The internal interrupt controller arranges these requests by priority and presents them one at a time to the CPU.

There are six external interrupt sources on the Am186EM and Am188EM microcontrollers—five maskable interrupt pins (INT4–INT0) and the non-maskable interrupt (NMI) pin. There are six internal interrupt sources that are not connected to external pins—three timers, two DMA channels, and the asynchronous serial port.

The Am186EM and Am188EM microcontrollers provide three interrupts that are not present on the 80C186 and 80C188 microcontrollers:

- INT4, an additional external interrupt pin that operates like the INT3–INT0 pins
- An internal watchdog timer interrupt
- An internal interrupt from the serial port

The INT4–INT0 interrupt request pins can be used as direct interrupt requests. If more inputs are needed, INT3–INT0 can also be cascaded with an 82C59A-compatible external interrupt control device. An external interrupt controller can be used as the system master by programming the internal interrupt controller to operate in slave mode. In all cases, nesting can be enabled that allows high priority interrupts to interrupt lower-priority interrupt service routines.

7.1.1 Definitions of Interrupt Terms

The following definitions cover some of the terminology that is used in describing the functionality of the interrupt controller. Table 7-1 contains information regarding the reserved interrupts.

7.1.1.1 Interrupt Type

An 8-bit interrupt type identifies each of the 256 possible interrupts.

Software exceptions, internal peripherals, and non-cascaded external interrupts supply the interrupt type through the internal interrupt controller.

Cascaded external interrupts and slave-mode external interrupts get the interrupt type from the external interrupt controller by means of interrupt acknowledge cycles on the bus.

7.1.1.2 Interrupt Vector Table

The interrupt vector table is a memory area of 1 Kbyte beginning at address 00000h that holds up to 256 four-byte address pointers containing the address for the interrupt service routine for each possible interrupt type. For each interrupt, an 8-bit interrupt type identifies the appropriate interrupt vector table entry.

Interrupts 00h to 1Fh are reserved. See Table 7-1.

The processor calculates the index to the interrupt vector table by shifting the interrupt type left 2 bits (multiplying by 4).

7.1.1.3 Maskable and Non-Maskable Interrupts

Interrupt types 08h through 1Fh are maskable. Of these, only 08h through 14h are actually in use (see Table 7-1). The maskable interrupts are enabled and disabled by the interrupt enable flag (IF) in the processor status flags, but the INT command can execute any interrupt regardless of the setting of IF.

Interrupt types 00h through 07h and all software interrupts (the INT instruction) are non-maskable. The non-maskable interrupts are not affected by the setting of the IF flag.

The Am186EM and Am188EM microcontrollers provide two methods for masking and unmasking the maskable interrupt sources. Each interrupt source has an interrupt control register that contains a mask bit specific to that interrupt. In addition, the Interrupt Mask register is provided as a single source to access all of the mask bits.

If the Interrupt Mask register is written while interrupts are enabled, it is possible that an interrupt could occur while the register is in an undefined state. This can cause interrupts to be accepted even though they were masked both before and after the write to the Interrupt Mask register. Therefore, the Interrupt Mask register should only be written when interrupts are disabled. Mask bits in the individual interrupt control registers can be written while interrupts are enabled, and there will be no erroneous interrupt operation.

7.1.1.4 Interrupt Enable Flag (IF)

The interrupt enable flag (IF) is part of the processor status flags (see section 2.1.1 on page 2-2). If IF is set to 1, maskable interrupts are enabled and can cause processor interrupts. (Individual maskable interrupts can still be disabled by means of the mask bit in each control register.)

If IF is set to 0, all maskable interrupts are disabled.

The IF flag does not affect the NMI or software exception interrupts (interrupt types 00h to 07h), and it does not affect the execution of any interrupt through the INT instruction.

7.1.1.5 Interrupt Mask Bit

Each of the interrupt control registers for the maskable interrupts contains a mask bit (MSK). If MSK is set to 1 for a particular interrupt, that interrupt is disabled regardless of the IF setting.

7.1.1.6 Interrupt Priority

The column titled *Overall Priority* in Table 7-1 shows the fundamental priority breakdown for the interrupts at power-on reset. The non-maskable interrupts 00h through 07h are always prioritized ahead of the maskable interrupts.

The maskable interrupts can be reprioritized by reconfiguring the PR2–PR0 bits in the interrupt control registers. The PR2–PR0 bits in all the maskable interrupts are set to priority level 7 at power-on reset.

7.1.1.7 Software Interrupts

Software interrupts can be initiated by the INT instruction. Any of the 256 possible interrupts can be initiated by the INT instruction. INT 21h causes an interrupt to the vector located at 00084h in the interrupt vector table. INT FFh causes an interrupt to the vector located at 003FCh in the interrupt vector table. Software interrupts are not maskable and are not affected by the setting of the IF flag.

7.1.1.8 Software Exceptions

A software exception interrupt occurs when an instruction causes an interrupt due to some condition in the processor. Interrupt types 00h, 01h, 03h, 04h, 05h, 06h, and 07h are software exception interrupts. Software exceptions are not maskable and are not affected by the setting of the IF flag.

Table 7-1 Am186EM and Am188EM Microcontroller Interrupt Types

Interrupt Name	Interrupt Type	Vector Table Address	EOI Type	Overall Priority	Related Instructions	Notes
Divide Error Exception	00h	00h	N/A	1	DIV, IDIV	1
Trace Interrupt	01h	04h	N/A	1A	All	2
Non-Maskable Interrupt (NMI)	02h	08h	N/A	1B		
Breakpoint Interrupt	03h	0Ch	N/A	1	INT 3	1
INTO Detected Overflow Exception	04h	10h	N/A	1	INTO	1
Array Bounds Exception	05h	14h	N/A	1	BOUND	1
Unused Opcode Exception	06h	18h	N/A	1	Undefined Opcodes	1
ESC Opcode Exception	07h	1Ch	N/A	1	ESC Opcodes	1, 3
Timer 0 Interrupt	08h	20h	08	2A		4, 5
Timer 1 Interrupt	12h	48h	08	2B		4, 5
Timer 2 Interrupt	13h	4Ch	08	2C		4, 5
Reserved for AMD Use	09h					
DMA 0 Interrupt	0Ah	28h	0A	3		5
DMA 1 Interrupt	0Bh	2Ch	0B	4		5
INT0 Interrupt	0Ch	30h	0C	5		
INT1 Interrupt	0Dh	34h	0D	6		
INT2 Interrupt	0Eh	38h	0E	7		
INT3 Interrupt	0Fh	3Ch	0F	8		
INT4 Interrupt	10h	40h	10	9		6
Watchdog Timer Interrupt	11h	44h	11	9		6
Asynchronous Serial Port Interrupt	14h	50h	14	9		6
Reserved for AMD Use	15h–1Fh					

Notes:

1. *Interrupts generated as a result of an instruction execution.*
2. *Trace is performed in the same manner as 80C186 and 80C188.*
3. *An ESC opcode causes a trap. This is part of the 80C186 and 80C188 co-processor interface, which is not supported on the Am186EM.*
4. *All three timers constitute one source of request to the interrupt controller. As such, they share the same priority level with respect to other interrupt sources. However, the timers have a defined priority order among themselves (2A>2B>2C).*
5. *The interrupt types of these sources are programmable in slave mode.*
6. *Not available in slave mode.*

7.1.2 Interrupt Conditions and Sequence

Interrupts are generally serviced as follows.

7.1.2.1 Non-Maskable Interrupts

Non-maskable interrupts—the trace interrupt, the NMI interrupt, and software interrupts [both user-defined (INT) and software exceptions]—are serviced regardless of the setting of the interrupt enable flag (IF) in the processor status flags.

7.1.2.2 Maskable Hardware Interrupts

In order for maskable hardware interrupt requests to be serviced, the IF flag must be set by the STI instruction, and the mask bit associated with each interrupt must be reset.

7.1.2.3 The Interrupt Request

When an interrupt is requested, the internal interrupt controller verifies that the interrupt is enabled and that there are no higher priority interrupt requests being serviced or pending. If the interrupt request is granted, the interrupt controller uses the interrupt type (see Table 7-1) to access a vector from the interrupt vector table.

Each interrupt type has a four-byte vector available in the interrupt vector table. The interrupt vector table is located in the 1024 bytes from 00000h to 003FFh. Each four-byte vector consists of a 16-bit offset (IP) value and a 16-bit segment (CS) value. The 8-bit interrupt type is shifted left 2 bit positions (multiplied by 4) to generate the index into the interrupt vector table.

7.1.2.4 Interrupt Servicing

A valid interrupt transfers execution to a new program location based on the vector in the interrupt vector table. The next instruction address (CS:IP) and the processor status flags are pushed onto the stack.

The interrupt enable flag (IF) is cleared after the processor status flags are pushed on the stack, disabling maskable interrupts during the interrupt service routine (ISR).

The segment:offset values from the interrupt vector table are loaded into the code segment (CS) and the instruction pointer (IP), and execution of the ISR begins.

7.1.2.5 Returning from the Interrupt

The interrupt return (IRET) instruction pops the processor status flags and the return address off the stack. Program execution resumes at the point where the interrupt occurred.

The interrupt enable flag (IF) is restored by the IRET instruction along with the rest of the processor status flags. If the IF flag was set before the interrupt was serviced, interrupts are re-enabled when the IRET is executed. If there are valid interrupts pending when the IRET is executed, the instruction at the return address is not executed. Instead, the new interrupt is serviced immediately.

If an ISR intends to permanently modify the value of any of the saved flags, it must modify the copy of the Processor Status Flags register that was pushed onto the stack.

7.1.3 Interrupt Priority

Table 7-1 shows the predefined types and overall priority structure for the Am186EM and Am188EM microcontrollers. Non-maskable interrupts (interrupt types 0–7) are always higher priority than maskable interrupts. Maskable interrupts have a programmable priority that can override the default priorities relative to one another.

The levels of interrupt priority are as follows:

- Interrupt priority for non-maskable interrupts and software interrupts
- Interrupt priority for maskable hardware interrupts

7.1.3.1 Non-Maskable Interrupts and Software Interrupt Priority

The non-maskable interrupts from 00h to 07h and software interrupts (INT instruction) always take priority over the maskable hardware interrupts. Within the non-maskable and software interrupts, the trace interrupt has the highest priority, followed by the NMI interrupt, followed by the remaining non-maskable and software interrupts.

After the trace interrupt and the NMI interrupt, the remaining software exceptions are mutually exclusive and can only occur one at a time, so there is no further priority breakdown.

7.1.3.2 Maskable Hardware Interrupt Priority

Beginning with interrupt type 8 (the Timer 0 interrupt), the maskable hardware interrupts have both an overall priority (see Table 7-1) and a programmable priority. The programmable priority is the primary priority for maskable hardware interrupts. The overall priority is the secondary priority for maskable hardware interrupts.

Since all maskable interrupts are set to a programmable priority of seven on reset, the overall priority of the interrupts determines the priority in which each interrupt is granted by the interrupt controller until programmable priorities are changed by reconfiguring the control registers.

The overall priority levels shown in Table 7-1 are not the same as the programmable priority level that is associated with each maskable hardware interrupt. Each of the maskable hardware interrupts has a programmable priority from zero to seven, with zero being the highest priority (see Table 7-3 on page 7-14).

For example, if the INT4–INT0 interrupts are all changed to programmable priority six and no other programmable priorities are changed from the reset value of seven, then the INT4–INT0 interrupts take precedence over all other maskable interrupts. (Within INT4–INT0, INT0 takes precedence over INT1, and INT1 takes precedence over INT2, etc., because of the underlying hierarchy of the overall priority.)

7.1.4 Software Exceptions, Traps, and NMI

The following predefined interrupts cannot be masked by programming.

7.1.4.1 Divide Error Exception (Interrupt Type 00h)

Generated when a DIV or IDIV instruction quotient cannot be expressed in the number of destination bits.

7.1.4.2 Trace Interrupt (Interrupt Type 01h)

If the trace flag (TF) in the Processor Status flags register is set, the trace interrupt is generated after most instructions. This interrupt allows programs to execute in single-step mode. The interrupt is not generated after prefix instructions like REP, instructions that modify segment registers like POP DS, or the WAIT instruction.

Taking the trace interrupt clears the TF bit after the processor status flags are pushed onto the stack. The IRET instruction at the end of the single step interrupt service routine restores the processor status flags (and the TF bit) and transfers control to the next instruction to be traced.

Trace mode is initiated by pushing the processor status flags onto the stack, setting the TF flag on the stack, and then popping the flags.

7.1.4.3 Non-Maskable Interrupt—NMI (Interrupt Type 02h)

The NMI pin provides an external interrupt source that is serviced regardless of the state of the IF (interrupt enable flag) bit. No external interrupt acknowledge sequence is performed for an NMI interrupt (see section 7.1.5). A typical use of NMI is to activate a power failure routine.

7.1.4.4 Breakpoint Interrupt (Interrupt Type 03h)

An interrupt caused by the 1-byte version of the INT instruction (INT3).

7.1.4.5 INTO Detected Overflow Exception (Interrupt Type 04h)

Generated by an INTO instruction if the OF bit is set in the Processor Status Flags (FLAGS) register.

7.1.4.6 Array BOUNDS Exception (Interrupt Type 05h)

Generated by a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

7.1.4.7 Unused Opcode Exception (Interrupt Type 06h)

Generated if execution is attempted on undefined opcodes.

7.1.4.8 ESC Opcode Exception (Interrupt Type 07h)

Generated if execution of ESC opcodes (D8h–DFh) is attempted. The microcontrollers do not check the escape opcode trap bit. The return address of this exception points to the ESC instruction that caused the exception. If a segment override prefix preceded the ESC instruction, the return address points to the segment override prefix.

Note: *All numeric coprocessor opcodes cause a trap. The Am186EM and Am188EM microcontrollers do not support the numeric coprocessor interface.*

7.1.5 Interrupt Acknowledge

Interrupts can be acknowledged in two different ways—the internal interrupt controller can provide the interrupt type or an external interrupt controller can provide the interrupt type. The processor requires the interrupt type as an index into the interrupt vector table.

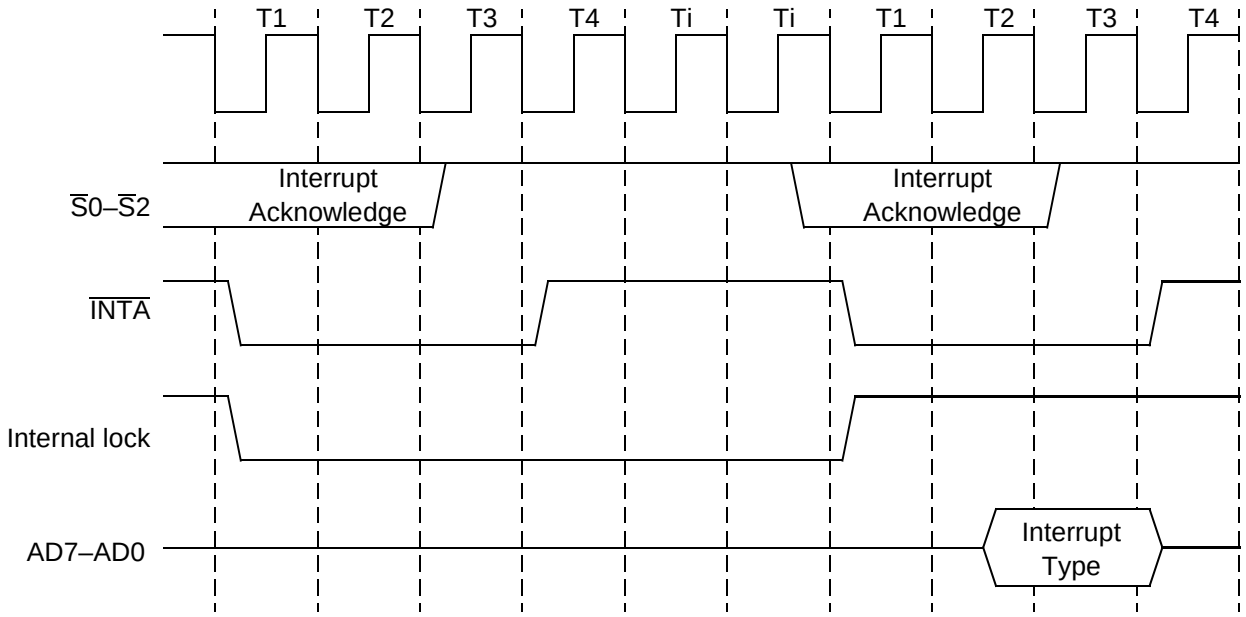
When the internal interrupt controller is supplying the interrupt type, no bus cycles are generated. The only external indication that an interrupt is being serviced is the processor reading the interrupt vector table.

When an external interrupt controller is supplying the interrupt type, the processor generates two interrupt acknowledge bus cycles (see Figure 7-1). The interrupt type is written to the AD7–AD0 lines by the external interrupt controller during the second bus cycle.

Interrupt acknowledge bus cycles have the following characteristics:

- The two interrupt acknowledge cycles are internally locked. (There is no $\overline{\text{LOCK}}$ pin on the Am186EM and Am188EM microcontrollers.)
- Two idle states are always inserted between the two cycles.
- Wait states are inserted if READY is not returned to the processor.

Figure 7-1 External Interrupt Acknowledge Bus Cycles



Notes:

1. ALE is active for each INTA cycle.

2. $\overline{RD}$ is inactive.

7.1.6 Interrupt Controller Reset Conditions

On reset, the interrupt controller performs the following nine actions:

1. All special fully nested mode (SFNM) bits are reset, implying fully nested mode.
2. All priority (PR) bits in the various control registers are set to 1. This places all sources at the lowest priority (level 7).
3. All level-triggered mode (LTM) bits are reset to 0, resulting in edge-triggered mode.
4. All interrupt in-service bits are reset to 0.
5. All interrupt request bits are reset to 0.
6. All mask (MSK) bits are set to 1. All interrupts are masked.
7. All cascade (C) bits are reset to 0 (non-cascade).
8. The interrupt priority mask is set to 7, allowing interrupts of all priorities.
9. The interrupt controller is initialized to master mode.

7.2 MASTER MODE OPERATION

This section describes master mode operation of the internal interrupt controller. See section 7.4 on page 7-28 for a description of slave mode operation.

Six pins are provided for external interrupt sources. One of these pins is NMI, the non-maskable interrupt. NMI is generally used for unusual events like power failure. The other five pins can be configured in any of the following ways:

- Fully nested mode—five interrupt lines with internally-generated interrupt types
- Cascade mode one—an interrupt line and interrupt acknowledge line pair with externally-generated interrupt types, plus three interrupt input lines with internally-generated types
- Cascade mode two—two pairs of interrupt and interrupt acknowledge lines with externally-generated interrupt types, and one interrupt input line (INT4) with internally-generated type

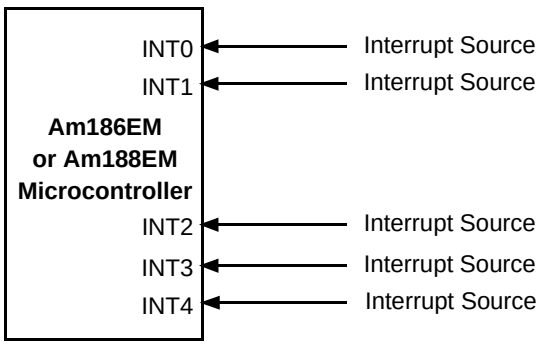
The basic modes of operation of the interrupt controller in master mode are similar to the 82C59A. The interrupt controller responds identically to internal interrupts in all three modes, the difference is only in the interpretation of function of the five external interrupt pins. The interrupt controller is set into one of these modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are fully nested mode, cascade mode, special fully nested mode, and polled mode.

7.2.1 Fully Nested Mode

In fully nested mode, five pins are used as direct interrupt requests as in Figure 7-2. The interrupt types for these five inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set for a higher priority interrupt, no interrupt is generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the in-service bit is set, no interrupt is generated by the interrupt controller. This allows interrupt service routines operating with interrupts enabled to be suspended only by interrupts of equal or higher priority than the in-service interrupt.

When an interrupt service routine is completed, the proper IS bit must be reset by writing the interrupt type to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. A write to the EOI register should be executed at the end of the interrupt service routine just before the return from interrupt instruction.

Figure 7-2 Fully Nested (Direct) Mode Interrupt Controller Connections



7.2.2 Cascade Mode

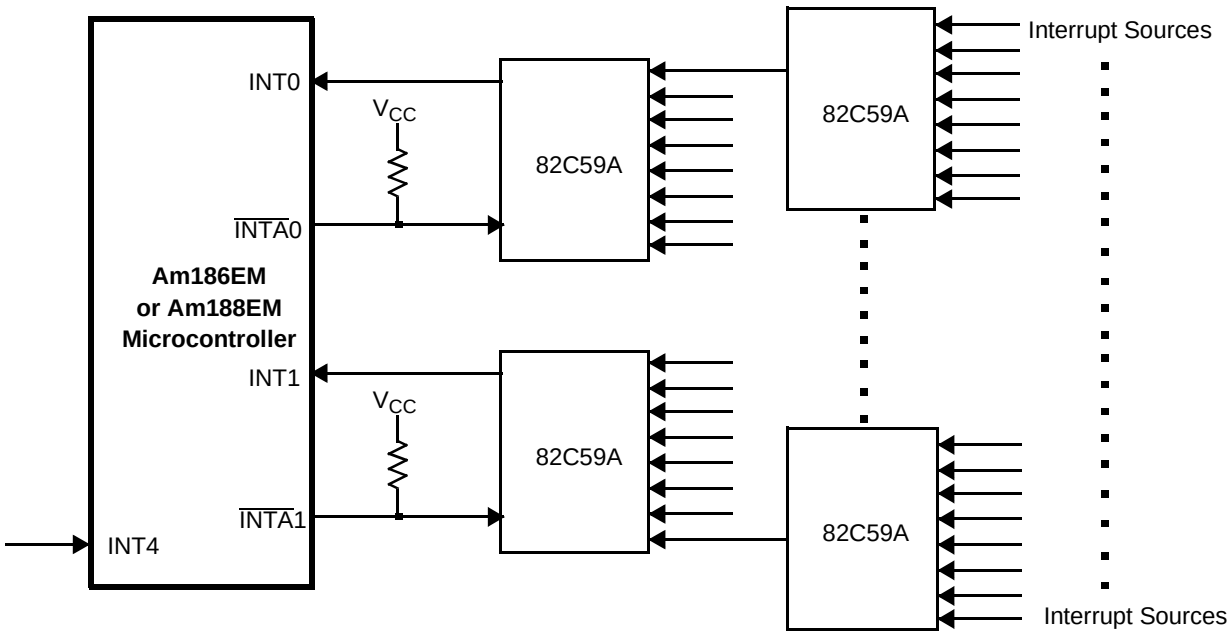
The Am186EM and Am188EM microcontrollers have five interrupt pins, two of which (INT2 and INT3) have dual functions. In fully nested mode, the five pins are used as direct interrupt inputs and the corresponding interrupt types are generated internally. In cascade mode, four of the five pins can be configured into interrupt input and dedicated acknowledge signal pairs. INT0 can be configured with interrupt acknowledge $\overline{\text{INTA}}$0 (INT2). INT1 can be configured with interrupt acknowledge $\overline{\text{INTA}}$1 (INT3).

External sources in cascade mode use externally generated interrupt types. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the type is read into the microcontroller on the second cycle (see section 7.1.5 on page 7-7). The capability to interface to one or two external 82C59A programmable interrupt controllers is provided when the inputs are configured in cascade mode.

Figure 7-3 shows the interconnection for cascade mode. INT0 is an interrupt input interfaced to one 82C59A, and INT2/$\overline{\text{INTA}}$0 serves as the dedicated interrupt acknowledge signal to that peripheral. INT1 and INT3/$\overline{\text{INTA}}$1 are also interfaced to an 82C59A. Each interrupt and acknowledge pair can be selectively placed in the cascade or non-cascade mode by programming the proper value into the INT0 and INT1 control registers. The dedicated acknowledge signals eliminate the need for external logic to generate $\overline{\text{INTA}}$ and device select signals.

Cascade mode provides the capability to serve up to 128 external interrupt sources through the use of external master and slave 82C59As. Three levels of priority are created, requiring priority resolution in the microcontroller interrupt controller, the master 82C59As, and the slave 82C59As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt (EOI) register writes must be issued by the program.

Figure 7-3 Cascade Mode Interrupt Controller Connections



7.2.3 Special Fully Nested Mode

Specially fully nested mode is entered by setting the SFNM bit in the INT0 or INT1 control registers. (See section 7.3.1 on page 7-13.) It enables complete nesting with external 82C59A masters or multiple interrupts from the same external interrupt pin when not in cascade mode. In this case, the ISRs must be re-entrant.

In fully nested mode, an interrupt request from an interrupt source is not recognized when the in-service bit for that source is set. In this case, if more than one interrupt source is connected to an external interrupt controller, all of the interrupts go through the same Am186EM or Am188EM microcontroller interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt is not recognized by the microcontroller until the in-service bit is reset.

In special fully nested mode, the microcontroller's interrupt controller allows the processor to take interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit continues to be set, however, to inhibit interrupts from other lower-priority Am186EM or Am188EM microcontroller interrupt sources.

In special fully nested mode with cascade mode, when a write is issued to the EOI register at the end of the interrupt service routine, software polling of the IS register in the external master 82C59A must determine if there is more than one IS bit set. If so, the IS bit in the microcontroller remains active and the next ISR is entered.

7.2.4 Operation in a Polled Environment

To allow reading of the Poll register information without setting the indicated in-service bit, the Am186EM and Am188EM microcontrollers provide a Poll Status register (Figure 7-15) in addition to the Poll register. Poll register information is duplicated in the Poll Status register, but the Poll Status register can be read without setting the associated in-service bit. These registers are located in two adjacent memory locations in the peripheral control block.

The interrupt controller can be used in polled mode if interrupts are not desired. When polling, interrupts are disabled and software polls the interrupt controller as required. The interrupt controller is polled by reading the Poll Status register (Figure 7-15). Bit 15 in the Poll Status register indicates to the processor that an interrupt of high enough priority is requesting service. Bits 4–0 indicate to the processor the interrupt type of the highest priority source requesting service. After determining that an interrupt is pending, software reads the Poll register (rather than the Poll Status register), which causes the in-service bit of the highest priority source to be set.

7.2.5 End-of-Interrupt Write to the EOI Register

A program must write to the EOI register to reset the in-service (IS) bit when an interrupt service routine is completed. There are two types of writes to the EOI register—specific EOI and non-specific EOI (see section 7.3.14 on page 7-27).

Non-specific EOI does not specify which IS bit is to be reset. Instead, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine.

Specific EOI requires the program to send the interrupt type to the interrupt controller to indicate the source IS bit that is to be reset. Specific reset is applicable when interrupt nesting is possible or when the highest priority IS bit that was set does not belong to the service routine in progress.

7.3 MASTER MODE INTERRUPT CONTROLLER REGISTERS

The interrupt controller registers for master mode are shown in Table 7-2. All the registers can be read and written unless otherwise specified.

Registers can be redefined in slave mode. See section 7.4 on page 7-28 for detailed information regarding slave mode register usage. On reset, the microcontroller is in master mode. Bit 14 of the relocation register (see Figure 4-2) must be set to initiate slave mode operation.

Table 7-2 Interrupt Controller Registers in Master Mode

Offset	Register Mnemonic	Register Name	Associated Pins	Comments
3Ah	I1CON	INT1 Control	INT1	
38h	I0CON	INT0 Control	INT0	
3Eh	I3CON	INT3 Control	INT3	
3Ch	I2CON	INT2 Control	INT2	
40h	I4CON	INT4 Control	INT4	
36h	DMA1CON	DMA1 Interrupt Control	DRQ1	
34h	DMA0CON	DMA0 Interrupt Control	DRQ0	
32h	TCUCON	Timer Interrupt Control	TMRIN1 TMRIN0 TMROUT1 TMROUT0	
42h	WDCON	Watchdog Timer Interrupt Control		
44h	SPICON	Serial Port Interrupt Control	TXD, RXD	
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request	INT4–INT0 DRQ1–DRQ0	Read-only register
2Ch	INSERV	In-Service	INT4–INT0 DRQ1–DRQ0	
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask	INT4–INT0 DRQ1–DRQ0	
26h	POLLST	Poll Status		Read-only register
24h	POLL	Poll		Read-only register
22h	EOI	End of Interrupt		Write-only register

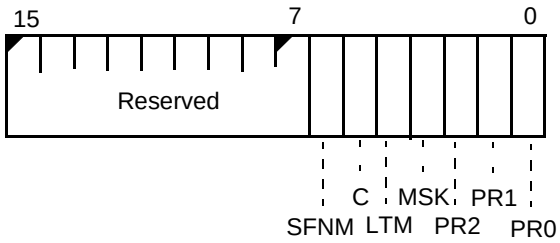
7.3.1 INT0 and INT1 Control Registers (I0CON, Offset 38h, I1CON, Offset 3Ah) (Master Mode)

The INT0 interrupt is assigned to interrupt type 0Ch. The INT1 interrupt is assigned to interrupt type 0Dh.

When cascade mode is enabled for INT0 by setting the C bit of I0CON to 1, the INT2 pin becomes $\overline{\text{INTA}}$0, the interrupt acknowledge for INT0.

When cascade mode is enabled for INT1 by setting the C bit of I1CON to 1, the INT3 pin becomes $\overline{\text{INTA}}$1, the interrupt acknowledge for INT1.

Figure 7-4 INT0 and INT1 Control Registers (I0CON, I1CON, offsets 38h and 3Ah)



The value of I0CON and I1CON at reset is 000Fh.

Bits 15–7: Reserved—Set to 0.

Bit 6: Special Fully Nested Mode (SFNM)—When set to 1, enables special fully nested mode.

Bit 5: Cascade Mode (C)—When set to 1, this bit enables cascade mode.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT0 or INT1 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT0 or INT1 as an active High, level-sensitive interrupt. A 0 in this bit configures INT0 or INT1 as a Low-to-High, edge-triggered interrupt. In either case, INT0 or INT1 must remain High until they are acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT0 or INT1 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT0 or INT1 from causing an interrupt. A 0 in this bit enables INT0 or INT1 interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT0 or INT1 relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

Table 7-3 Priority Level

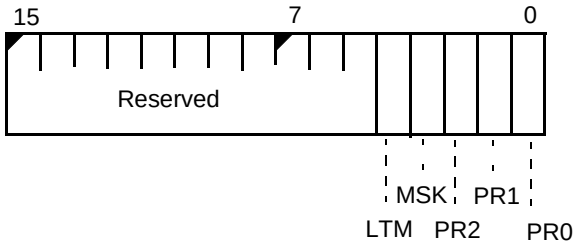
Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

7.3.2 INT2 and INT3 Control Registers (I2CON, Offset 3Ch, I3CON, Offset 3Eh) (Master Mode)

The INT2 interrupt is assigned to interrupt type OEh. The INT3 interrupt is assigned to interrupt type 0Fh.

The INT2 and INT3 pins can be configured as interrupt acknowledge pins $\overline{\text{INTA}}$0 and $\overline{\text{INTA}}$1 when cascade mode is implemented.

Figure 7-5 INT2 and INT3 Control Registers (I2CON, I3CON, offsets 3Ch and 3Eh)



The value of I2CON and I3CON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT2 or INT3 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT2 or INT3 as an active High, level-sensitive interrupt. A 0 in this bit configures INT2 or INT3 as a Low-to-High, edge-triggered interrupt. In either case, INT2 or INT3 must remain High until they are acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT2 or INT3 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT2 or INT3 from causing an interrupt. A 0 in this bit enables INT2 or INT3 interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

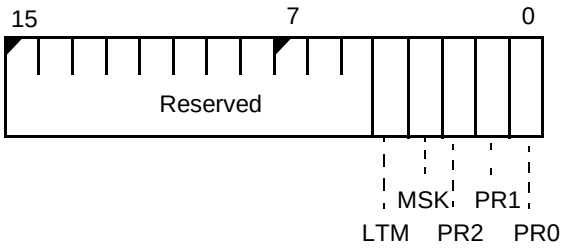
Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT2 or INT3 relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.3.3 INT4 Control Register (I4CON, Offset 40h) (Master Mode)

The Am186EM and Am188EM microcontrollers provide INT4, an additional external interrupt pin. This input behaves like INT3–INT0 on the 80C186/188 microcontroller with the exception that INT4 is only intended for use as a nested-mode interrupt source.

This interrupt is assigned to interrupt type 10h. The Interrupt 4 Control register (see Figure 7-6) controls the operation of the INT4 signal.

Figure 7-6 INT4 Control Register (I4CON, offset 40h)



The value of I4CON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT4 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT4 as an active High, level-sensitive interrupt. A 0 in this bit configures INT4 as a Low-to-High, edge-triggered interrupt. In either case, INT4 must remain High until it is acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT4 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT4 from causing an interrupt. A 0 in this bit enables INT4 interrupts.

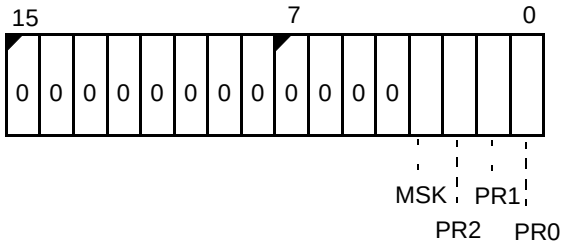
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority (PR)—This field determines the priority of INT4 relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.3.4 Timer and DMA Interrupt Control Registers (TCUCON, Offset 32h, DMA0CON, Offset 34h, DMA1CON, Offset 36h) (Master Mode)

The three timer interrupts are assigned to interrupt type 08h, 12h, and 13h. All three timer interrupts are configured through TCUCON, offset 32h. The DMA0 interrupt is assigned to interrupt type 0Ah. The DMA1 interrupt is assigned to interrupt type 0Bh.

Figure 7-7 Timer/DMA Interrupt Control Registers (TCUCON, DMA0CON, DMA1CON, offsets 32h, 34h, and 36h)



The value of TCUCON, DMA0CON, and DMA1CON at reset is 000Fh.

Bits 15–4: Reserved—Set to 0.

Bit 3: Interrupt Mask (MSK)—This bit determines whether the corresponding signal can generate an interrupt. A 1 masks this interrupt source. A 0 enables the corresponding interrupt.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

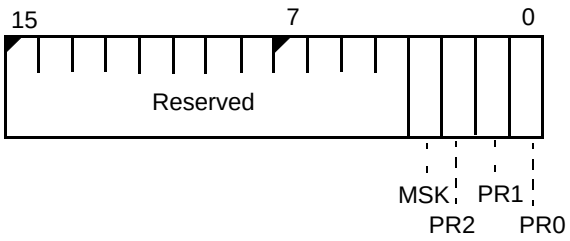
Bits 2–0: Priority Level (PR2–PR0)—Sets the priority level for its corresponding source. See Table 7-3 on page 7-14.

7.3.5 Watchdog Timer Interrupt Control Register (WDCON, Offset 42h) (Master Mode)

The Am186EM and Am188EM microcontrollers provide an additional on-chip interrupt source, the watchdog timer. This timer is constructed from existing 80C186 microcontroller pins. It is implemented by connecting the TMROUT1 output to an additional internal interrupt to create the watchdog timer interrupt. This interrupt is assigned to interrupt type 11h. The control register format is shown in Figure 7-8.

The systems programmer should program the timer (see section 8.2.2 on page 8-3) and then program the interrupt pin.

Figure 7-8 Watchdog Timer Interrupt Control Register (WDCON, offset 42h)



The value of WDCON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Reserved—*Must* be set to 0 to ensure proper operation of the Am186EM and Am188EM microcontrollers.

Bit 3: Mask (MSK)—This bit determines whether the watchdog timer can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the watchdog timer from causing an interrupt. A 0 in this bit enables watchdog timer interrupts.

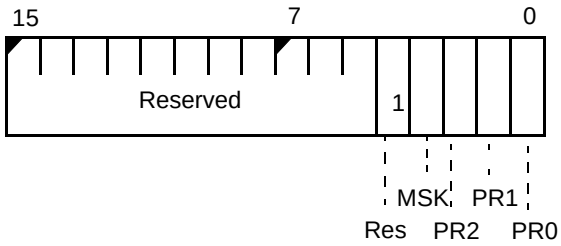
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority (PR)—This field determines the priority of the watchdog timer relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.3.6 Serial Port Interrupt Control Register (SPICON, Offset 44h) (Master Mode)

The Serial Port Interrupt Control register controls the operation of the asynchronous serial port interrupt source (SPI, bit 10 in the Interrupt Request register). This interrupt is assigned to interrupt type 14h. The control register format is shown in Figure 7-9.

Figure 7-9 Serial Port Interrupt Control Register (SPICON, offset 44h)



The value of SPICON at reset is 001Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Reserved—Set to 1.

Bit 3: Mask (MSK)—This bit determines whether the serial port can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the serial port from causing an interrupt. A 0 in this bit enables serial port interrupts.

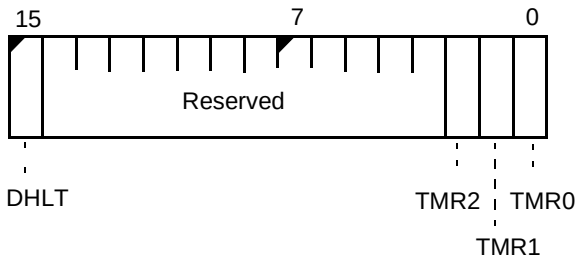
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority (PR2–PR0)—This field determines the priority of the serial port relative to the other interrupt signals. After a reset, the priority is 7. See Table 7-3 on page 7-14.

7.3.7 Interrupt Status Register (INTSTS, Offset 30h) (Master Mode)

The Interrupt Status (INTSTS) register indicates the interrupt request status of the three timers.

Figure 7-10 Interrupt Status Register (INTSTS, offset 30h)



Bit 15: DMA Halt (DHLT)—When set to 1, halts any DMA activity. This pin is automatically set to 1 when non-maskable interrupts occur and is reset when an IRET instruction is executed. Time-critical software, such as interrupt handlers, can modify this bit directly to inhibit DMA transfers. Because of the function of this register as an interrupt request register for the timers, the DHLT bit should not be modified by software when timer interrupts are enabled.

Bits 14–3: Reserved

Bits 2–0: Timer Interrupt Request (TMR2–TMR0)—When set to 1, these bits indicate that the corresponding timer has an interrupt request pending. (Note that the timer TMR bit in the REQST register is the OR of these timer interrupt requests.)

7.3.8 Interrupt Request Register (REQST, Offset 2Eh) (Master Mode)

The hardware interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the REQST register is shown in Figure 7-11.

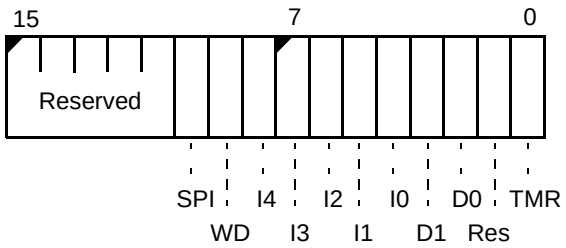
The Am186EM and Am188EM microcontrollers define three new bits to report the state of INT4, the Watchdog Timer, and the asynchronous serial port.

For internal interrupts (SPI, WD, D1, D0, and TMR), the corresponding bit is set to 1 when the device requests an interrupt. The bit is reset during the internally generated interrupt acknowledge.

For INT4–INT0 external interrupts, the corresponding bit (I4–I0) reflects the current value of the external signal. The device must hold this signal High until the interrupt is serviced.

Generally the interrupt service routine signals the external device to remove the interrupt request.

Figure 7-11 Interrupt Request Register (REQST, offset 2Eh)



The REQST register is undefined on reset.

Bits 15–11: Reserved

Bit 10: Serial Port Interrupt Request (SPI)—This bit indicates the interrupt state of the serial port. If enabled, the SPI bit is the logical OR of all possible serial port interrupt sources (THRE, RDR, BRKI, FER, PER, and OER status bits).

Bit 9: Watchdog Timer Interrupt Request (WD)—When this bit is set to 1, the Watchdog Timer has an interrupt pending.

Bits 8–4: Interrupt Requests (I4–I0)—When set to 1, the corresponding INT pin has an interrupt pending (i.e., when INT0 is pending, I0 is set). These bits reflect the status of the external pin.

Bits 3–2: DMA Channel Interrupt Request (D1–D0)—When set to 1, the corresponding DMA channel has an interrupt pending.

Bit 1: Reserved

Bit 0: Timer Interrupt Request (TMR)—This bit indicates the state of the timer interrupts. This bit is the logical OR of the timer interrupt requests. When set to a 1, this bit indicates that the timer control unit has an interrupt pending.

The Interrupt Status register indicates the specific timer that is requesting an interrupt. See section 7.3.7.

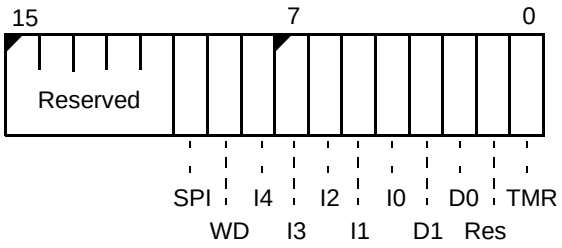
7.3.9 In-Service Register (INSERV, Offset 2Ch) (Master Mode)

The Am186EM and Am188EM microcontrollers define three new bits to report the in-service state of INT4, the Virtual Watchdog Timer, and the asynchronous serial port. The format of the modified In-Service register is shown in Figure 7-12.

The bits in the INSERV register are set by the interrupt controller when the interrupt is taken. Each bit in the register is cleared by writing the corresponding interrupt type to the End-of-Interrupt (EOI) register. See Table 7-1 on page 7-3.

When an in-service bit is set, the microcontroller will not generate an interrupt request for the associated source, preventing an interrupt from interrupting itself if interrupts are enabled in the ISR. Special fully nested mode allows the INT1–INT0 requests to circumvent this restriction for the INT0 and INT1 sources.

Figure 7-12 In-Service Register (INSERV, offset 2Ch)



The INSERV register is set to 0000h on reset.

Bits 15–11: Reserved

Bit 10: Serial Port Interrupt In-Service (SPI)—This bit indicates the in-service state of the asynchronous serial port.

Bit 9: Watchdog Timer Interrupt In-Service (WD)—This bit indicates the in-service state of the Watchdog Timer.

Bits 8–4: Interrupt In-Service (I4–I0)—These bits indicate the in-service state of the corresponding INT pin.

Bits 3–2: DMA Channel Interrupt In-Service (D1–D0)—These bits indicate the in-service state of the corresponding DMA channel.

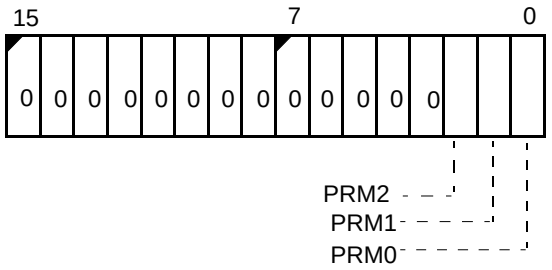
Bit 1: Reserved

Bit 0: Timer Interrupt In-Service (TMR)—This bit indicates the state of the in-service timer interrupts. This bit is the logical OR of all the timer interrupt status bits. When set to a 1, this bit indicates that the corresponding timer interrupt status bit is in-service.

7.3.10 Priority Mask Register (PRIMSK, Offset 2Ah) (Master Mode)

The Priority Mask (PRIMSK) register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt.

Figure 7-13 Priority Mask Register (PRIMSK, offset 2Ah)



The value of PRIMSK at reset is 0007h.

Bits 15–3: Reserved—Set to 0.

Bits 2–0: Priority Field Mask (PRM2–PRM0)—This field determines the minimum priority that is required in order for a maskable interrupt source to generate an interrupt. Maskable interrupts with programmable priority values that are numerically higher than this field are masked. The possible values are zero (000b) to seven (111b).

A value of seven (111b) allows all interrupt sources that are not masked to generate interrupts. A value of five (101b) allows only unmasked interrupt sources with a programmable priority of zero to five (000b to 101b) to generate interrupts.

Table 7-4 Priority Level

Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

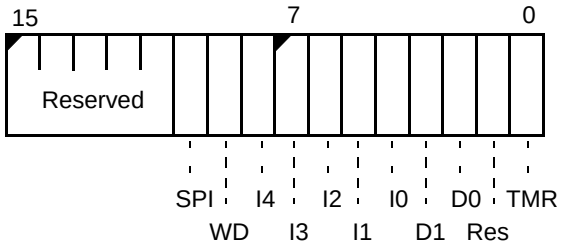
7.3.11 Interrupt Mask Register (IMASK, Offset 28h) (Master Mode)

The Am186EM and Am188EM microcontrollers define three new bits to report the mask state of the INT4 Control, Watchdog Timer Interrupt Control, and Serial Port Interrupt Control registers.

The Interrupt Mask (IMASK) register is a read/write register. Programming a bit in the IMASK register has the effect of programming the MSK bit in the associated control register. The format of the IMASK register is shown in Figure 7-14.

Do not write to the interrupt mask register while interrupts are enabled. To modify mask bits while interrupts are enabled, use the individual interrupt control registers.

Figure 7-14 Interrupt Mask Register (IMASK, offset 28h)



The IMASK register is set to 07FDh on reset.

Bits 15–11: Reserved

Bit 10: Serial Port Interrupt Mask (SPI)— When set to 1, this bit indicates that the asynchronous serial port interrupt is masked.

Bit 9: Virtual Watchdog Timer Interrupt Mask (WD)—When set to 1, this bit indicates that the Watchdog Timer interrupt is masked.

Bits 8–4: Interrupt Mask (I4–I0)—When set to 1, an I4–I0 bit indicates that the corresponding interrupt is masked.

Bits 3–2: DMA Channel Interrupt Masks (D1–D0)—When set to 1, a D1–D0 bit indicates that the corresponding DMA channel interrupt is masked.

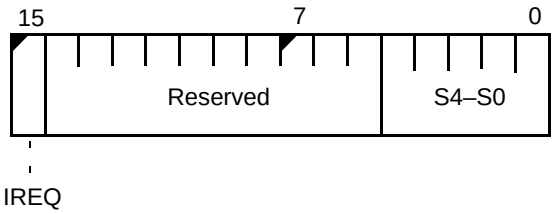
Bit 1: Reserved

Bit 0: Timer Interrupt Mask (TMR)—When set to 1, this bit indicates that interrupt requests from the timer control unit are masked.

7.3.12 Poll Status Register (POLLST, Offset 26h) (Master Mode)

The Poll Status (POLLST) register mirrors the current state of the Poll register. The POLLST register can be read without affecting the current interrupt request. But when the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register.

Figure 7-15 Poll Status Register (POLLST, offset 26h)



Bit 15: Interrupt Request (IREQ)—Set to 1 if an interrupt is pending. When this bit is set to 1, the S4–S0 field contains valid data.

Bits 14–5: Reserved—Set to 0.

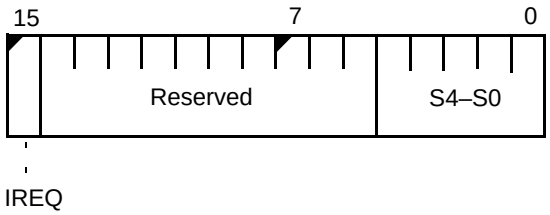
Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt.

7.3.13 Poll Register (POLL, Offset 24h) (Master Mode)

When the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register.

The Poll Status register mirrors the current state of the Poll register, but the Poll Status register can be read without affecting the current interrupt request.

Figure 7-16 Poll Register (POLL, offset 24h)



Bit 15: Interrupt Request (IREQ)—Set to 1 if an interrupt is pending. When this bit is set to 1, the S4–S0 field contains valid data.

Bits 14–5: Reserved—Set to 0.

Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt. Reading the Poll register acknowledges the highest priority pending interrupt and enables the next interrupt to advance into the register.

Although the IS bit is set, the interrupt service routine does not begin execution automatically. The application software must execute the appropriate ISR.

7.3.14 End-of-Interrupt Register (EOI, Offset 22h) (Master Mode)

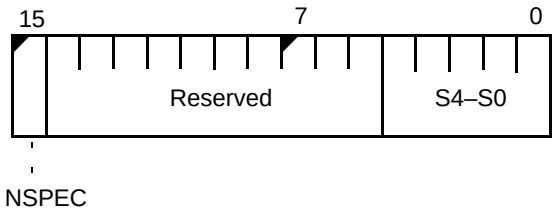
The End-of-Interrupt (EOI) register is a write-only register. The in-service flags in the In-Service register (see section 7.3.9 on page 7-22) are reset by writing to the EOI register. Before executing the IRET instruction that ends an interrupt service routine (ISR), the ISR should write to the EOI register to reset the IS bit for the interrupt.

The specific EOI reset is the most secure method to use for resetting IS bits. Figure 7-17 shows example code for a specific EOI reset. See Table 7-1 on page 7-3 for specific EOI values.

Figure 7-17 Example EOI Assembly Code

```
            ...
            ...                     ;ISR code
            ...
            mov dx, EOI_ADDR
exit:       mov ax,int_type         ;load the interrupt type in ax
            out dx,ax               ;write the interrupt type to EOI
            popa
            iret                    ;return from interrupt
```

Figure 7-18 End-of-Interrupt Register (EOI, offset 22h)



Bit 15: Non-Specific EOI (NSPEC)—The NSPEC bit determines the type of EOI command. When written as a 1, NSPEC indicates non-specific EOI. When written as a 0, NSPEC indicates the specific EOI interrupt type in S4–S0.

Bits 14–5: Reserved

Bits 4–0: Source EOI Type (S4–S0)—Specifies the EOI type of the interrupt that is currently being processed. See Table 7-1 on page 7-3.

7.4 SLAVE MODE OPERATION

When slave mode is used, the microcontroller's internal interrupt controller is used as a slave controller to an external master interrupt controller. The internal interrupts are monitored by the internal interrupt controller, while the external controller functions as the system master interrupt controller.

On reset, the microcontroller is in master mode. To activate slave mode operation, bit 14 of the relocation register must be set (see Figure 4-2 on page 4-4).

Because of pin limitations caused by the need to interface to an external 82C59A master, the internal interrupt controller does not accept external inputs. However, there are enough interrupt controller inputs (internally) to dedicate one to each timer. In slave mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The INT4, watchdog timer, and serial port interrupts are not available in slave mode. In slave mode, each peripheral must be assigned a unique priority to ensure proper interrupt controller operation. The programmer must assign correct priorities and initialize interrupt control registers before enabling interrupts.

7.4.1 Slave Mode Interrupt Nesting

Slave mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

7.4.2 Slave Mode Interrupt Controller Registers

The Interrupt Controller Registers for slave mode are shown in Table 7-5. All registers can be read and written, unless specified otherwise.

Table 7-5 Interrupt Controller Registers in Slave Mode

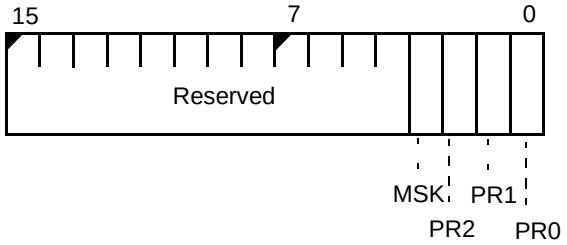
Offset	Register Mnemonic	Register Name	Affected Pins	Comments
3Ah	T2INTCON	Timer 2 Interrupt Control		Interrupt Type XXXXX101
38h	T1INTCON	Timer 1 Interrupt Control	TMRIN1 TMROUT1	Interrupt Type XXXXX100
36h	DMA1CON	DMA 1 Interrupt Control		Interrupt Type XXXXX011
34h	DMA0CON	DMA 0 Interrupt Control		Interrupt Type XXXXX010
32h	T0INTCON	Timer 0 Interrupt Control	TMRIN0 TMROUT0	Interrupt Type XXXXX000
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request		Read Only
2Ch	INSERV	In-Service		Read Only
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask		
22h	EOI	Specific EOI		Write Only
20h	INTVEC	Interrupt Vector		

7.4.3 Timer and DMA Interrupt Control Registers (T0INTCON, Offset 32h, T1INTCON, Offset 38h, T2INTCON, Offset 3Ah, DMA0CON, Offset 34h, DMA1CON, Offset 36h) (Slave Mode)

In slave mode, there are three separate registers for the three timers. In master mode, all three timers are masked and prioritized in one register, TCUCON.

In slave mode, the two DMA control registers retain their functionality and addressing from master mode.

Figure 7-19 Timer and DMA Interrupt Control Registers (T0INTCON, T1INTCON, T2INTCON, DMA0CON, DMA1CON, offsets 32h, 38h, 3Ah, 34h, and 36h)



These registers are set to 000Fh on reset.

Bits 15–4: Reserved—Set to 0.

Bit 3: Mask (MSK)—This bit determines whether the interrupt source can cause an interrupt. A 1 in this bit masks the interrupt source, preventing the source from causing an interrupt. A 0 in this bit enables interrupts from the source.

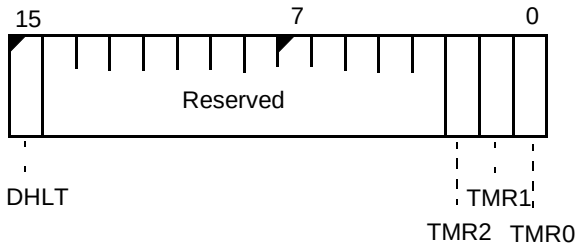
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.4.8 on page 7-34.

Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of the interrupt source relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.4.4 Interrupt Status Register (INTSTS, Offset 30h) (Slave Mode)

The Interrupt Status register controls DMA activity when non-maskable interrupts occur and indicates the current interrupt status of the three timers.

Figure 7-20 Interrupt Status Register (INTSTS, offset 30h)



The INTSTS register is set to 0000h on reset.

Bit 15: DMA Halt (DHLT)—When set to 1, halts any DMA activity. Automatically set to 1 when non-maskable interrupts occur and reset when an IRET instruction is executed.

Bits 14–3: Reserved

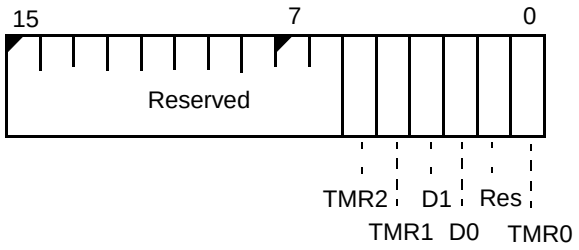
Bits 2–0: Timer Interrupt Request (TMR2–TMR0)—When set to 1, indicates the corresponding timer has an interrupt request pending.

7.4.5 Interrupt Request Register (REQST, Offset 2Eh) (Slave Mode)

The internal interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the Interrupt Request register is shown in Figure 7-21.

For internal interrupts (D1, D0, TMR2, TMR1, and TMR0), the corresponding bit is set to 1 when the device requests an interrupt. The bit is reset during the internally generated interrupt acknowledge.

Figure 7-21 Interrupt Request Register (REQST, offset 2Eh)



The REQST register is set to 0000h on reset.

Bits 15–6: Reserved

Bits 5–4: Timer 2/Timer 1 Interrupt Request (TMR2–TMR1)—When set to 1, these bits indicate the state of any interrupt requests from the associated timer.

Bits 3–2: DMA Channel Interrupt Request (D1–D0)—When set to 1, D1–D0 indicate that the corresponding DMA channel has an interrupt pending.

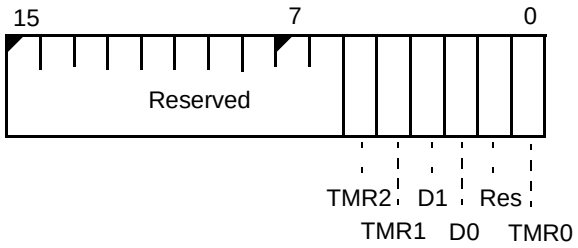
Bit 1: Reserved

Bit 0: Timer 0 Interrupt Request (TMR0)—When set to 1, this bit indicates the state of an interrupt request from Timer 0.

7.4.6 In-Service Register (INSERV, Offset 2Ch) (Slave Mode)

The format of the In-Service register is shown in Figure 7-22. The bits in the In-Service register are set by the interrupt controller when the interrupt is taken. The in-service bits are cleared by writing to the End-of-Interrupt (EOI) register.

Figure 7-22 In-Service Register (INSERV, offset 2Ch)



The INSERV register is set to 0000h on reset.

Bits 15–6: Reserved

Bits 5–4: Timer 2/Timer 1 Interrupt In-Service (TMR2–TMR1)—When set to 1, these bits indicate that the corresponding timer interrupt is currently being serviced.

Bits 3–2: DMA Channel Interrupt In-Service (D1–D0)—When set to 1, the corresponding DMA channel is currently being serviced.

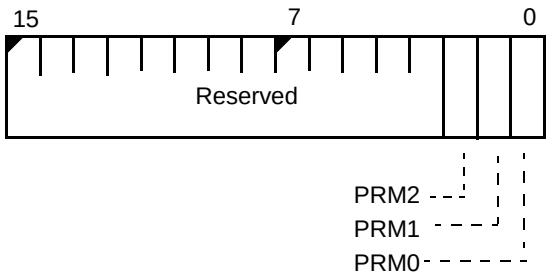
Bit 1: Reserved

Bit 0: Timer 0 Interrupt In-Service (TMR0)—When set to 1, this bit indicates Timer 0 is currently being serviced.

7.4.7 Priority Mask Register (PRIMSK, Offset 2Ah) (Slave Mode)

The format of the Priority Mask register is shown in Figure 7-23. The Priority Mask register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt.

Figure 7-23 Priority Mask Register (PRIMSK, offset 2Ah)



The value of the PRIMSK register at reset is 0007h.

Bits 15–3: Reserved

Bits 2–0: Priority Field Mask (PRM2–PRM0)—This field determines the minimum priority which is required in order for a maskable interrupt source to generate an interrupt.

A value of seven (111b) allows all interrupt sources that are not masked to generate interrupts. A value of five (101b) allows only unmasked interrupt sources with a programmable priority of zero to five (000b to 101b) to generate interrupts.

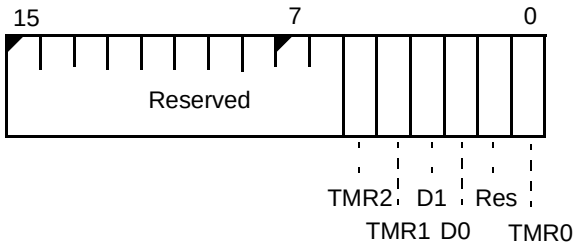
Table 7-6 Priority Level

Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

7.4.8 Interrupt Mask Register (IMASK, Offset 28h) (Slave Mode)

The format of the Interrupt Mask register is shown in Figure 7-24. The Interrupt Mask register is a read/write register. Programming a bit in the Interrupt Mask register has the effect of programming the MSK bit in the associated control register.

Figure 7-24 Interrupt Mask Register (IMASK, offset 28h)



The IMASK register is set to 003Dh on reset.

Bits 15–6: Reserved

Bits 5–4: Timer 2/Timer 1 Interrupt Mask (TMR2–TMR1)—These bits indicate the state of the mask bit of the Timer Interrupt Control register and when set to a 1, indicate which source has its interrupt requests masked.

Bits 3–2: DMA Channel Interrupt Mask (D1–D0)—These bits indicate the state of the mask bits of the corresponding DMA control register.

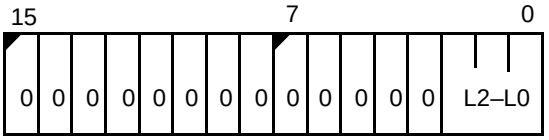
Bit 1: Reserved

Bit 0: Timer 0 Interrupt Mask (TMR0)—This bit indicates the state of the mask bit of the Timer Interrupt Control register and when set to a 1, indicates Timer 0 has its interrupt request masked.

7.4.9 Specific End-of-Interrupt Register (EOI, Offset 22h) (Slave Mode)

In slave mode, a write to the EOI register resets an in-service bit of a specific priority. The user supplies a three-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22h.

Figure 7-25 Specific End-of-Interrupt Register (EOI, offset 22h)



The EOI register is undefined on reset.

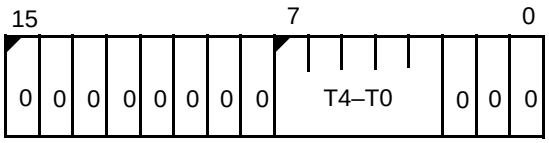
Bits 15–3: Reserved—Write as 0.

Bits 2–0: Interrupt Type (L2–L0)—Encoded value indicating the priority of the IS (interrupt service) bit to be reset. Writes to these bits cause an EOI to be issued for the interrupt type in slave mode. Write-only register.

7.4.10 Interrupt Vector Register (INTVEC, Offset 20h) (Slave Mode)

Vector generation in slave mode is exactly like that of an 8259A or 82C59A slave. The interrupt controller generates an 8-bit interrupt type that the CPU shifts left two bits (multiplies by four) to generate an offset into the interrupt vector table.

Figure 7-26 Interrupt Vector Register (INTVEC, offset 20h)



The INTVEC register is undefined on reset.

Bits 15–8: Reserved—Read as 0.

Bits 7–3: Interrupt Type (T4–T0)—Sets the five most significant bits of the interrupt types for the internal interrupt type. The interrupt controller itself provides the lower three bits of the interrupt type, as determined by the priority level of the interrupt request. See Table 7-5 on page 7-15.

Bits 2–0: Reserved—Read as 0.